Claims:

- 1. A stack configured in a nonvolatile memory to store parameter values.
- 2. The stack of claim 1 wherein the nonvolatile memory includes a pair of blocks that are erased independently.
- 3. The stack of claim 2 wherein valid parameter values are stored in a first block of the pair of blocks and a second block is erased.
- 4. The stack of claim 3 wherein valid parameter values are stored in the second block of the pair of blocks and the first block is erased.
- 5. The stack of claim 1 further including a register to store an offset value used to generate an address for words in the nonvolatile memory.
- 6. The stack of claim 1 further including a smart stack controller to dynamically determine a number of blocks used in the stack.
- 7. The stack of claim 1 further including a smart stack controller to distribute write cycles across multiple blocks of the nonvolatile memory.
- 8. The stack of claim 1 wherein the nonvolatile memory includes polymer memory devices.
- 9. The stack of claim 8 wherein the polymer memory devices are plastic memory devices.
- 10. The stack of claim 8 wherein the polymer memory devices are resistive change polymer memory devices.

- 11. A nonvolatile stack to store parameter values in words of a nonvolatile memory.
- 12. The nonvolatile stack of claim 11, wherein a memory pool in at least first and second blocks of the nonvolatile memory are sized to balance cycling and data retention capabilities with a write specification.
- 13. The nonvolatile stack of claim 11, further including a stack controller to distribute write cycles across multiple blocks of the nonvolatile memory.
- 14. The nonvolatile stack of claim 11, wherein the nonvolatile memory maps a received address to determine memory blocks to be written.

15. A storage device, comprising:

a nonvolatile memory having multiple blocks in a dynamic block swapped architecture, wherein a pair of blocks are configured to provide a first stack that stores parameter values.

- 16. The storage device of claim 15 further including a smart stack controller to distribute write cycles across the multiple blocks.
- 17. The storage device of claim 15 further including a smart stack controller to dynamically determine which blocks from the multiple blocks are used in the first stack.
- 18. The storage device of claim 15 wherein a second pair of blocks are instantiated in the storage device to configure a second stack.
- 19. The storage device of claim 15 wherein multiple stacks are instantiated in the storage device with two blocks shared among the multiple stacks.

20. A computer system, comprising:

first and second antennas:

- a transceiver coupled to the first and second antennas;
- a processor coupled to the transceiver; and
- a nonvolatile memory coupled to the processor to provide a nonvolatile stack to store parameter values.
- 21. The computer system of claim 20 wherein the nonvolatile memory includes first and second blocks that are configured to form the nonvolatile stack and are erased independently.
- 22. The computer system of claim 20 further including a register to store an offset value used to generate an address for words in the nonvolatile memory.
- 23. The computer system of claim 20 where the nonvolatile stack includes polymer memory devices.
- 24. The computer system of claim 20 where the nonvolatile stack includes Chalcogenide memory devices.
- 25. The computer system of claim 20 where the nonvolatile stack includes microelectromechanical (MEM) memory devices.

26. A method comprising:

configuring a stack in first and second blocks of a nonvolatile memory; and pushing data onto the stack.

27. The method of claim 26, further including:

receiving an address from a processor that is translated by the nonvolatile memory to an address location for storing parameter values in the first and second blocks.

28. The method of claim 26, further including:

writing data to a second memory block of the nonvolatile memory when a first memory block is full.

29. The method of claim 28, further including:

erasing the first memory block when the data stored in the first memory block is entirely invalid.

30. The method of claim 26, further including:

scanning for a last valid data entry by confining searches to one memory block for a last stored value within the nonvolatile memory.

31. The method of claim 26, further including:

reading the data at an address determined based on register contents and a processor-supplied address.